Project_desc

Course: Fundamentals of Cloud Computing

Project Title: Real-Time Data Ingestion & Deployment on AWS Cloud for Airline Vendor Ecosystem

Duration: 2–3 Weeks

Level: Intermediate

Platform: AWS Cloud

Tech Stack: AWS EC2, SQS, Lambda, DynamoDB, Docker, IAM, ECR, .NET

Project Background

TELEMAX, a global airline management company, is expanding its operations to underserved markets. As a part of their digital transformation strategy, TELEMAX is building a cloud-native system to accept real-time data orders from vendors and store the data securely for future analysis.

In this project, students are required to consult TELEMAX on deploying an effective and scalable AWS-based solution to ingest real-time data, process it through a decoupled architecture, and store it in a NoSQL database for further analytics and optimization.

Project Objective

Design and deploy a microservice-based, real-time data ingestion pipeline on AWS using SQS, Lambda, DynamoDB, EC2, and Docker. A key deliverable includes packaging a .NET program into a Docker container and deploying it to Amazon ECR.

Deliverables

- Successfully configured AWS architecture
- Functional Lambda function triggered via SQS
- Dockerized .NET program deployed to ECR

- Documented Word file with screenshots and architecture explanation
- Final presentation

Project Tasks

Task 1: Create a Virtual Machine (EC2 Instance)

- Launch an Ubuntu/Linux EC2 instance on AWS.
- Ensure appropriate security group rules for SSH access.

Task 2: Create and Test an SQS Queue

- Create an AWS SQS (Standard) Queue.
- Test with dummy vendor data messages.

Task 3: Install Docker on EC2 Instance

- Install Docker on the EC2 instance.
- Test Docker installation with a sample container.

Task 4: Create IAM Role with ECR Permissions

- Create an IAM role with limited permissions.
- Ensure it allows access to Amazon ECR for pushing Docker images.

Task 5: Attach IAM Role to EC2

- Attach the IAM role created in Task 4 to the EC2 instance.

Task 6: Create a DynamoDB Table

- Create a DynamoDB table to store incoming vendor data.
- Define primary keys and necessary attributes.

Task 7: Create Lambda Function to Store Data in DynamoDB

- Develop and configure a Lambda function.
- Function should accept data and write it to the DynamoDB table.

Task 8: Connect Lambda to SQS Queue

- Trigger the Lambda function whenever a message is received in the SQS queue.
- Ensure smooth integration and test the flow.

Task 9: Final Report and Presentation

- Prepare a detailed project report with architecture diagrams and step-by-step screenshots.
- Present the project to peers/instructor, explaining each step and challenges faced.

Learning Outcomes

Upon successful completion, students will:

- Understand how to use AWS SQS, Lambda, DynamoDB, EC2, IAM, and Docker in tandem.
- Learn to containerize applications and deploy them to AWS ECR.
- Build hands-on experience in developing serverless and decoupled architectures.
- Learn best practices in AWS IAM, security, and resource provisioning.